

Einstiegsaufgaben:

Für die Teilnahme müssen 2 von den 3 Einstiegsaufgabe erfolgreich in C, C++ oder Java bearbeitet worden sein.

Die Abgabe der Programme erfolgt als Kurz-Demonstration am ersten
Praktikumstermin im Sommersemester 2009, Di 14.4.2009, 13:15 Uhr, Raum N-907.

Bringen Sie Ihre Programme bitte als kommentierten Sourcecode und als ausführbares Programm mit (z.B. auf Memory Stick oder CD).

Erstellen Sie zu dem Programm noch eine DIN-A4 Seite Beschreibung zu dem Programm in der Sie die Bedienung und die Programmstruktur kurz erläutern; (in etwa so was wie eine kurze Users Reference und Programmers Reference).

- Aufgabe I: Digitales Filter
Aufgabe II: Differenzialgleichung
Aufgabe III: Abstand zwischen Polygonen

Einstiegsaufgabe I: Digitales Filter

Erstellen Sie ein Programm, welches Messwerte $y[i] = f(x[i])$ aus einer Textdatei einliest, filtert und dann in eine zweite Textdatei ausgibt, je ein Wert als ASCII Text pro Zeile.

Für einen Messwert i , wird aus mehreren Funktionswerten (z.B: $y[i-1]$, $y[i]$, $y[i+1]$) ein gewichteter Mittelwert gebildet und als neuer Wert $z[i]$ abgespeichert:

$$z[i] = s * (p2*y[i-2] + p1*y[i-1] + p0*y[i] + q1*y[i+1] + q2*y[i+2])$$

Auf diese Weise wird mit allen verfügbaren Werten der Funktion verfahren und daraus die gefilterte Funktion aus z -Werten berechnet und abgespeichert.

Die Anzahl der Gewichtung-Parameter p, q soll maximal 7 sein und die Gewichtungswerte $p3, \dots, q3$ und s sollen vom Benutzer eingegeben werden können.

Bsp 1: symmetrischer 7-Punkte Moving Average Filter Kern:

$$p3 = p2 = p1 = p0 = q1 = q2 = q3 = 1.0 \text{ und } s=1/7$$

Einstiegsaufgabe II: Differenzialgleichung

Erstellen Sie ein Programm, welches **T**-Zeitschritte lang ein System von drei Differenzialgleichungen numerisch mit der sog. Euler-Vorwärts / Tangentenmethode berechnet und die Funktionswerte in eine Textdatei ausgibt.

Das Differenzialgleichungssystem (DGL- \mathcal{L}) besteht aus drei Gleichungen für die zeitlichen Ableitungen **dX**, **dY**, **dZ** der drei Größen **X**, **Y**, **Z**, und den konstanten Koeffizienten **r,s,q**:

$$\begin{aligned} dX(t) &= -s \cdot X(t) + s \cdot Y(t) \\ dY(t) &= r \cdot X(t) - Y(t) - X(t) \cdot Z(t) \\ dZ(t) &= -q \cdot Z(t) + X(t) \cdot Y(t) \end{aligned}$$

Mit der Tangentenmethode nach Euler ergeben sich die nächsten Werte jeweils zu:
X(t+1) = X(t) + dt*dX(t), **Y(t+1)=Y(t)+dt*dY(t)**, **Z(t+1)= Z(t)+dt*dZ(t)**

Schreiben Sie in jedem Schritt die Werte **t**, **X(t)**, **Y(t)**, **Z(t)**, **dX(t)**, **dY(t)**, **dZ(t)** in eine Textdatei: eine Zeile pro Zeitschritt, die einzelnen Werte durch Leerzeichen oder Tab getrennt. Erlauben Sie, dass die Koeffizienten des Systems **r,s,q** die Startwerte **X(0)**, **Y(0)**, **Z(0)**, und die Parameter **dt** und **T** vom Benutzer eingegeben werden können aber nicht müssen.

Default Werte: **X(0) = Y(0) = Z(0) = 0.726**, **r = 10.0**, **s=17.0**, **q=2.67**, **dt=0.01**, **T=10000**

Einstiegsaufgabe III: Abstand zwischen Polygonen

Erstellen Sie ein Programm, welches den kürzesten Abstand **r** zwischen einem Dreieck **D** und einem Viereck **V** in der Ebene berechnet.

Beide Polygone sind durch Ihre Eckpunkte bestimmt.

Das Dreieck **D** durch **(D1x,D1y)**, **(D2x,D2y)**, **(D3x,D3y)**.

Das Viereck **V** durch **(V1x,V1y)**, **(V2x,V2y)**, **(V3x,V3y)**, **(V4x, V4y)**.

Das Programm soll die benötigten Werte aus einer Datei oder von der Tastatur einlesen und daraus den kürzesten Abstand **r** berechnen und ausgegeben.

Zur Vereinfachung kann davon ausgegangen werden, dass die Polygone sich nicht schneiden, also keine der Verbindungslinien zwischen den Punkten eine andere Verbindungslinie schneidet.